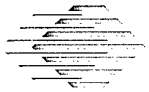


Carnegie Mellon
Software Engineering Institute

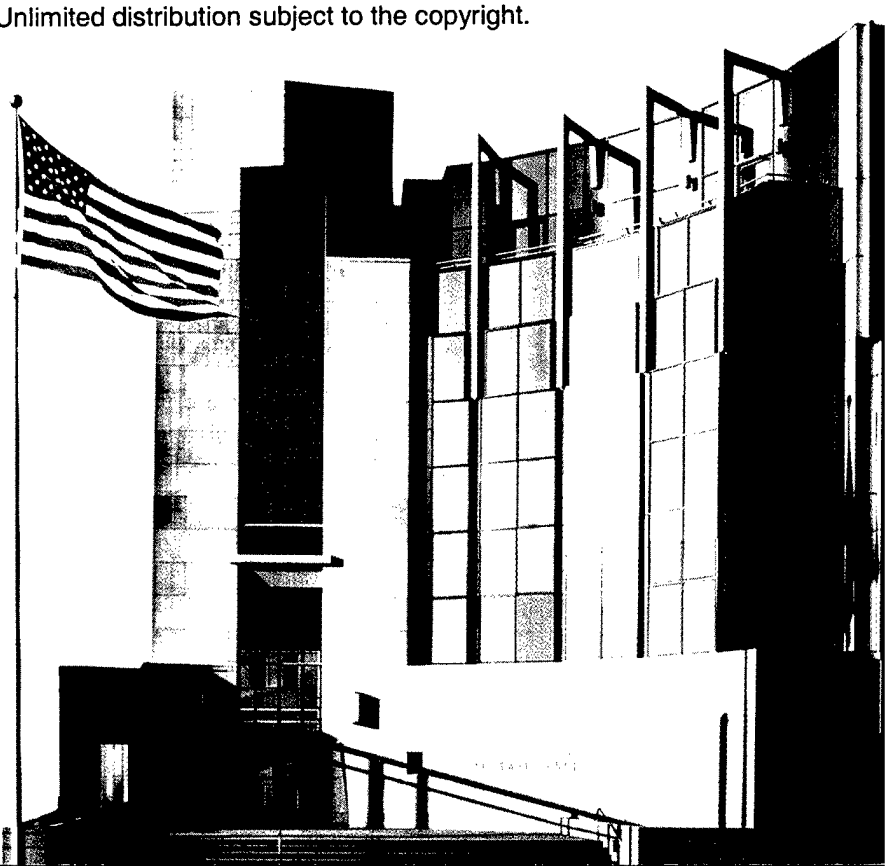
Use of the Architecture Tradeoff Analysis Method[SM] (ATAM[SM]) in Source Selection of Software-Intensive Systems

John K. Bergey
Matthew J. Fisher
Lawrence G. Jones

June 2002

Architecture Tradeoff Analysis Initiative

Technical Note
CMU/SEI-2002-TN-010



Carnegie Mellon University does not discriminate and Carnegie Mellon University is required not to discriminate in admission, employment, or administration of its programs or activities on the basis of race, color, national origin, sex or handicap in violation of Title VI of the Civil Rights Act of 1964, Title IX of the Educational Amendments of 1972 and Section 504 of the Rehabilitation Act of 1973 or other federal, state, or local laws or executive orders.

In addition, Carnegie Mellon University does not discriminate in admission, employment or administration of its programs on the basis of religion, creed, ancestry, belief, age, veteran status, sexual orientation or in violation of federal, state, or local laws or executive orders. However, in the judgment of the Carnegie Mellon Human Relations Commission, the Department of Defense policy of "Don't ask, don't tell, don't pursue" excludes openly gay, lesbian and bisexual students from receiving ROTC scholarships or serving in the military. Nevertheless, all ROTC classes at Carnegie Mellon University are available to all students.

Inquiries concerning application of these statements should be directed to the Provost, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-6684 or the Vice President for Enrollment, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-2056.

Obtain general information about Carnegie Mellon University by calling (412) 268-2000.

Use of the Architecture Tradeoff Analysis Method[SM] (ATAM[SM]) in Source Selection of Software-Intensive Systems

John K. Bergey
Matthew J. Fisher
Lawrence G. Jones

June 2002

Architecture Tradeoff Analysis Initiative

Technical Note
CMU/SEI-2002-TN-010

Contents

List of Figures

About the Technical Note Series on Software Architecture Evaluation in the Department of Defense

The Product Line Systems Program is publishing a series of technical notes designed to condense knowledge about evaluation practices for software architecture into a concise and usable form for the Department of Defense (DoD) acquisition manager and practitioner. This series is a companion to the Software Engineering Institute (SEI) series on product line acquisition and business practices [Bergey 99].

Each technical note in the series will focus on the use of software architecture evaluation and, in particular, on applying the SEI's architecture tradeoff analysis technology in the Department of Defense and government organizations. Our objective is to provide practical guidance on ways to integrate sound evaluation practices for software architecture into their acquisitions. This series of technical notes will lay down a conceptual foundation for the DoD's evaluation practices for software architecture.

Abstract

Software architecture is critical to the quality of a software-intensive system. For an acquisition organization, such as the Department of Defense (DoD), the ability to evaluate software architectures as early as possible in an acquisition can have a favorable impact on the delivered system. This technical note explains the role of software architecture evaluation in a source selection and describes the contractual elements that are needed to support its use. The note then briefly describes the Architecture Tradeoff Analysis Method[SM] (ATAM[SM]) and provides an example that shows how to apply this method in a source selection. The example includes sample contractual language that an acquirer can adapt to meet specific acquisition needs.

1 Introduction

The software architecture of a system significantly influences the overall functionality, performance, and other qualities of that system. The use of software architecture evaluations early in a system acquisition can help mitigate many of the technical risks associated with system development, thereby improving the ability of an organization to achieve the stated system objectives.[1] In an acquisition context, these evaluations provide the acquirer with a proactive means of

- gaining early visibility into critical tradeoffs and design decisions that will drive the entire system-development effort
- determining if a system being proposed by a supplier will satisfy its desired system quality attributes before the system is actually built

This technical note discusses how an organization can use the Architecture Tradeoff Analysis Method[SM] (ATAM [SM]) for software architecture evaluation during source selection in a software-intensive system acquisition. It describes the contents of typical solicitation packages, such as a request for proposal (RFP), to illustrate where the evaluation method may be incorporated. It outlines briefly the steps of the ATAM and provides sample acquisition language (here defined as wording for a solicitation package, associated statement of work [SOW], and a system specification). This technical note complements an earlier technical note on applying the ATAM to a system acquisition in the early stages of software architecture development, following contract award [Bergey 01]. For readers familiar with this earlier work, the primary difference in this technical note is that it describes how to apply the ATAM during source selection. The resultant changes are found in Sections 3 and 4 of this technical note and the appendices.

[1] Fisher, M. "Software Architecture Awareness and Training for Software Practitioners." Pittsburgh, PA: U.S. Army CECOM Course (June 1998).

[SM] Architecture Tradeoff Analysis Method and ATAM are service marks of Carnegie Mellon University.

2 Software Architecture in System Acquisition

2.1 The Role of Software Architecture

Software is pervasive in many modern systems. Software is also the root cause of many of today's system problems. Moreover, the quality and longevity of a software-intensive system is determined by its software architecture.

> *The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them [Bass 98].*

The software architecture is the foundation for any software system. It represents the earliest design decisions that are both the most difficult to get right and the hardest to change downstream. The software architecture will allow or preclude nearly all of the system's quality attributes. The quality attributes (such as modifiability, performance predictability, security, availability, interoperability, and usability) are all largely pre-cast when the software architecture has been established. No amount of later tuning and implementation tactics will compensate for the sins of a poorly constructed software architecture. Experience has shown that an unsuitable software architecture will eventually precipitate some sort of disaster on a project. Disaster may mean failure to meet performance goals, failure to interoperate as needed, and/or inordinate sustainment costs, among other problems.

If functionality were all that mattered, any monolithic software architecture would do, but other things–namely the quality attributes–do matter. Time and care must be invested in the software architecture up front because making changes later is extremely costly and often impossible. The software architecture should then guide how implementation is carried out. Throughout the development process, the software architecture must play a role that is both prescriptive and descriptive. Even in an incremental acquisition and development approach, the core system and software architectural decisions that support the important quality attribute goals for the system must come first, and then they can be enhanced in future increments or spirals. An architecture-centric approach is key to the development of systems that meet both their functional and quality goals.

While the emphasis in this technical note is on software architecture and the so-called "non-functional" or quality requirements, it is still vital that the acquirer specify the functional requirements for a system in a complete solicitation package.

2.2 Software Architecture Evaluation in Systems Acquisition

2.2.1 Phases of an Acquisition

In this technical note, we consider the activities corresponding to three phases of an acquisition: pre-award, award, and post-award [Bergey 99]. These activities are illustrated in Figure 1. Software architecture evaluation can potentially play a role in all these phases to help lower the risks associated with an acquisition.

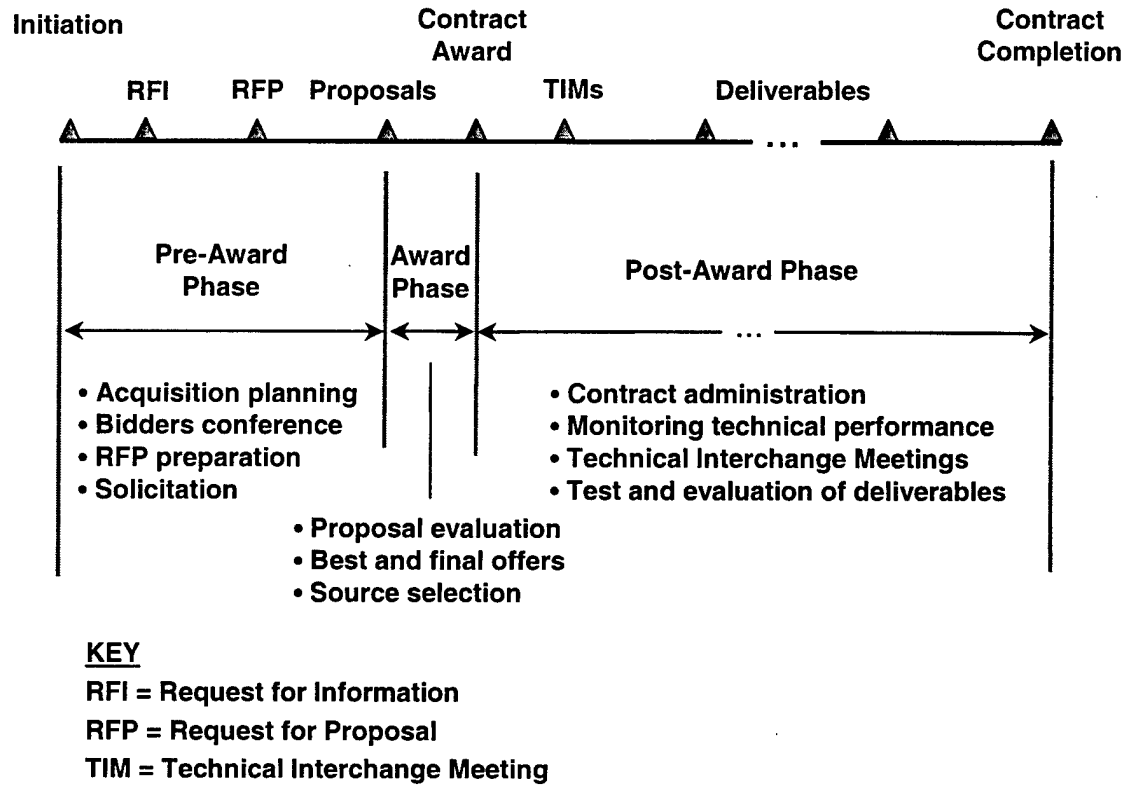


Figure 1: Phases of an Acquisition

To use software architecture evaluation in either the award phase (e.g., source selections) or the post-award phase (e.g., contract management), the solicitation package must contain the criteria for proposal and product evaluation and include the software architecture evaluation method to be used. This essential groundwork is laid during the pre-award phase.

During the award phase, software architecture evaluations can be used to evaluate suppliers' overall approaches to satisfying system requirements, to assess the strengths and weaknesses of proposed software architectures, to identify risks to the program, and to assess each offeror's ability to participate in or conduct software architecture evaluations.

During the post-award phase, software architecture evaluations can be used for contract management by enabling acquirers to evaluate both supplier and product performance.

2.2.2 Pre-Award and Award Phase for a System-Development Contract

Acquisition planning should be an ongoing activity throughout the acquisition. However, sufficient acquisition planning must precede the solicitation process to generate and validate the foundational product requirements (e.g., functionality and quality requirements such as performance).

In the pre-award phase, a solicitation package is developed. It tells potential suppliers what the requirements[2] of the acquisition are, how to prepare their proposals, how proposals will be evaluated, and when to submit their proposals [Cooper 02]. Solicitation packages take various forms and are referred to differently. However, they all have the same characteristics noted here. We will use the common term *request for proposals* (RFP) to refer to solicitation packages.

The RFP typically contains Sections A through M. These sections provide information that must be distributed to potential suppliers (i.e., offerors). Depending upon the acquiring organization's policies and processes, the sections may be incorporated in different ways. Most RFPs, however, contain the same type of information.

The RFP and eventual contract language should

- address the acquisition requirements of the project
- comply with regulations, policies, and other guidance
- clearly describe product requirements in terms of functionality, performance, and quality
- protect the interests of both the acquirer (buyer) and the supplier (contractor)

The contents of an RFP and the resulting contract depend largely upon the acquirer's knowledge and objectives for the acquisition. In the context of this technical note, the RFP sections must include the requirement for software architecture evaluation. As a result, in this technical note we are interested in Sections C, L, and M, which are shown in Figure 2. We will review the contents of these sections to demonstrate some of the considerations that are needed to incorporate a software architecture evaluation into an acquisition.

[2] The term *requirements* encompasses all requirements of the acquisition, including product requirements, where the term *product* may mean a specific system or service [Cooper 99].

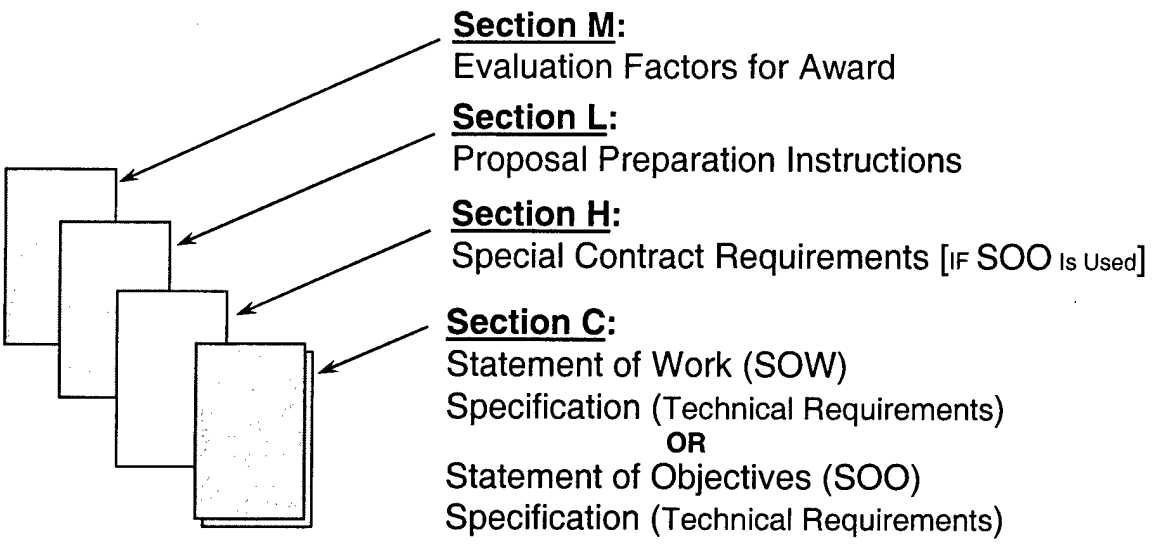


Figure 2: Typical Contents of Requests for Proposals (RFPs)

2.2.2.1 Section C

Section C normally contains supplier work requirements in the form of a statement of work (SOW) along with product requirements such as a system performance specification (containing functional and quality requirements[3]). If a software architecture evaluation is to be required, both the SOW and the product requirements must specify the particular method (such as the ATAM) as well as how the software architecture evaluation method will be used and implemented in the acquisition. This information must be integrated and compatible with other acquisition requirements that are part of the RFP.

Statement of Work (SOW)

The statement of work (SOW) describes what the supplier must accomplish. In terms of any evaluation method, the SOW describes which evaluation steps are the supplier's responsibilities. The software architecture evaluation steps in the SOW must be consistent with the overall acquisition. In addition, the SOW should indicate if certain evaluation steps are to be performed jointly by the acquirer and the potential system supplier.

Sometimes an acquisition organization will elect (or is required) to include a statement of objectives (SOO) in the RFP instead of a SOW. In these cases, the contract language that would traditionally be included in the SOW (to describe the requirements for software architecture evaluation) should be included under Section H (Special Contract Requirements) of the RFP.

---

[3] If an SOO is used, the technical requirements document (TRD) contains the specific system quality requirements.

System Requirements

A system specification typically has two main sections of interest. Section 1 specifies functional and quality requirements for the system. Here, quality requirements refer to the quality attributes of the system and their respective characterizations. Modifiability, reliability, and security are examples of the types of system quality attributes that may be considered. For example, if reliability is a required quality attribute, a characterization might be that the system must meet a specific mean time between failure (MTBF) requirement. Eliciting the quality attributes of primary interest as well as their characterizations is part of the ATAM.

Among other things, Section 2 of the system specification describes the software architecture evaluation methods (such as the ATAM) that the supplier must use to evaluate the software architecture during the post-award phase of the acquisition. The evaluation results will be the basis for determining if the software architecture can support the satisfaction of the requirements in Section 1 of the system specification.

2.2.2.2 Section L

Section L (Proposal Preparation Instructions) describes what offerors should address in their proposals and the response that is required. Typically, the acquirer would ask the potential suppliers for responses in several volumes (such as a technical volume, past performance volume, management volume, and cost volume). The acquirer has great latitude in specifying the contents of these volumes. In the technical volume, an acquirer may ask potential suppliers to describe their proposed approach for implementing the software architecture requirements and performing a software architecture evaluation. In the past performance volume, an acquirer may ask suppliers to describe previous work on software architecture development and evaluation.

2.2.2.3 Section M

Section M (Evaluation Factors for Award) tells potential suppliers how their proposals will be evaluated. This typically includes specifying

- what areas (i.e., factors and subfactors) of the offeror's proposed approach are to be evaluated as part of proposal evaluation
- the specific criteria to be used for judging the offeror's proposed approach to meeting the RFP/contract requirements for these factors and subfactors

To incorporate software architecture evaluation, Section M must specify how it will relate to the factors and subfactors. In addition, it must specify the criteria to be used in judging the offeror's approach to satisfying the RFP/contract software architecture and software architecture evaluation requirements.

It is important to emphasize that all RFP sections must be consistent with each other. For example, Section M must include the specific criteria to evaluate only those RFP responses that correspond to the requested areas identified in Section L.

From a contracting officer's perspective, releasing the RFP defines the official beginning of the solicitation. After the solicitation period formally closes on the specified date, source selection commences with a proposal evaluation and ends with a contract award. The software architecture evaluation results can be included as part of the proposal evaluations as long as the proposal evaluation criteria explicitly accommodate this.

3 The Architecture Tradeoff Analysis Method (ATAM) and Source Selection

In this section, we discuss a particular method for software architecture evaluation, the Architecture Tradeoff Analysis Method (ATAM), and how it can be applied in source selection. We first present the standard approach for conducting an ATAM evaluation followed by a discussion of considerations for its use in source selection.

3.1 The Architecture Tradeoff Analysis Method

The ATAM is a useful technique for analyzing and evaluating software architectures. The SEI developed and refined this method over the past six years [Kazman 00]. It not only can be used to evaluate architectural decisions against specific quality attributes, it also allows engineering tradeoffs to be made among possibly conflicting system quality goals. In this way, the ATAM evaluation can detect areas of potential risk in meeting quality goals within the architecture of a complex software-intensive system. Clements, Kazman, and Klein provide details and examples of the ATAM (and other software architecture evaluation methods) [Clements 02a].

The ATAM has several advantages. It can be done early, quickly, and inexpensively. The method involves project decision makers, other stakeholders (including managers, developers, maintainers, testers, re-users, end users, and customers), and a software architecture evaluation team. These groups collaborate to determine the critical quality attributes of the system and effectively evaluate the consequences of architectural decisions in light of specified quality attributes and business goals. The method helps to ensure that the right questions are asked to uncover

- risks – software architecture decisions that might create future problems in some quality attribute
- sensitivity points – properties of one or more components (and/or component relationships) that are critical for achieving a particular quality attribute response (That is, a slight change in a property can make a significant difference in a quality attribute.)
- tradeoffs – decisions affecting more than one quality attribute

There are nine specific steps in the basic ATAM evaluation that fall into four general types of activities: presentation, investigation and analysis, testing, and reporting.

3.1.1 Presentation

Step 1. Present the ATAM. The method is described to the assembled stakeholders (typically customer representatives, the architect or software architecture team, user representatives, maintainers, administrators, managers, testers, integrators, etc.).

Step 2. Present business drivers. The project manager describes the business goals that are motivating the development effort and hence the primary software architecture drivers (e.g., broad availability, time to market, high security).

Step 3. Present software architecture. The architect describes the proposed software architecture, focusing on how it addresses the business drivers.

3.1.2 Investigation and Analysis

Step 4. Identify architectural approaches. Architectural approaches are identified by the architect, but they are not analyzed.

Step 5. Generate quality attribute utility tree. The quality attributes that comprise system "utility" (performance, reliability, security, modifiability, etc.) are elicited. These are specified down to the level of scenarios, annotated with stimuli and responses, and prioritized. A scenario is a short statement describing an interaction of a stakeholder with the system. Scenarios provide a vehicle for making vague qualities concrete.[4]

Step 6. Analyze architectural approaches. Based upon the high-priority factors identified in Step 5, the architectural approaches that address those factors are elicited and analyzed. For example, an architectural approach aimed at meeting performance goals will be subjected to a performance analysis. During this step, software architecture risks, sensitivity points, and tradeoff points are identified.

3.1.3 Testing

Step 7. Brainstorm and prioritize scenarios. Based upon the example scenarios generated in the utility tree step, a larger set of scenarios is elicited from the entire group of stakeholders. This set of scenarios is prioritized via a voting process involving the entire stakeholder group.

Step 8. Analyze architectural approaches. This step reiterates Step 6; but here, the highly ranked scenarios from Step 7 are considered to be test cases for software architecture approaches determined thus far. These test case scenarios may uncover additional software architecture approaches, risks, sensitivity points, and tradeoff points, which are then documented.

---

[4] Examples of quality attributes and scenarios may be found in Appendix C.

3.1.4 Reporting

Step 9. Present results. Based upon the information collected during the ATAM evaluation (styles, scenarios, attribute-specific questions, the utility tree, risks, sensitivity points, tradeoffs), the evaluation team presents its findings to the assembled stakeholders and details this information along with any proposed mitigation strategies in a written report.

It is important to have the roles and responsibilities understood before incorporating the ATAM or any software evaluation method in an acquisition. The sections of the RFP and, ultimately, the contract must clearly reflect this understanding. In the next section, we discuss how to use the ATAM during source selection.

3.2 Using the ATAM in Source Selection

There are two important factors in any source selection that limit how the ATAM may be applied. First, to maintain fairness (and avoid protests), the communications between the acquirer and all offerors must be formal and consistent. Second, the solicitation process does not naturally lend itself to iteration.

Given these limitations, we suggest the following six steps for applying the ATAM in source selection.

Step 1. Document business goals and quality requirements.

In this step the acquirer determines the business or mission goals and system quality requirements, and then documents them as part of the RFP. This information will establish a fair and consistent basis for potential suppliers to respond to the RFP and for proposal evaluators to make their recommendations. It also provides a crucial basis for the design of the system and evaluation of the software architecture. The more detailed this information is, the more useful it will be to suppliers. This step can be achieved by including the following information in the RFP:

- the business drivers motivating the development effort
- the key quality attributes desired in the system
- a set of scenarios that characterize the quality attributes in operational terms
- requirements for documenting[5] the software architecture to permit the demonstration and evaluation of how it supports the required quality attributes

The acquirer might find it useful to consider applying the principles of some of the ATAM steps when preparing this information. Of particular use might be

[5] In their book, Clements, Kazman, and Klein provide a detailed treatment of documenting software architecture [Clements 02b].

- ATAM Step 2 (result: business drivers)
- ATAM Step 5 (result: quality attribute tree)
- ATAM Step 7 (result: prioritized scenarios)

The RFP must include the requirement for potential suppliers to conduct and demonstrate their execution of an ATAM evaluation. Just as with a typical ATAM, it is important to have key system stakeholders involved in these steps. Because there will be no participation by the actual system developers prior to releasing the RFP, it is important to have experts participate to represent interests common to software developers and maintainers. Additionally, by specifying the requirements of software architecture documentation, the acquirer establishes the basis for offerors to conduct their ATAM evaluations.[6]

The acquirer may pursue different strategies regarding how to treat potential suppliers' knowledge (or lack of it) of the ATAM. Since software architecture evaluations are not common in DoD acquisitions, the acquiring organization might consider scheduling an ATAM tutorial as part of a bidders' conference.[7] This conference would provide all offerors with a basic understanding of the ATAM to help them better understand the RFP's requirements for software architecture analysis. Conducting an ATAM tutorial would also provide a suitable forum for answering any questions the participants may have about the prescribed software architecture analysis method.[8] Alternately, the acquirer may choose not to do this and to consider each supplier's existing capability to perform an ATAM evaluation as a "past performance" rating factor.

Step 2. Develop software architecture.

In this step, a potential supplier develops (or adapts) and documents a software architecture to meet the requirements of the RFP. The acquirer must allow sufficient time for suppliers to execute this step. This time period might be relatively short in the case of a mature application domain where suppliers might be expected to have existing software architectures (or where a market survey confirms this existing architecture). Even in this case the acquirer must allow sufficient time for suppliers to make modifications to their existing software architectures. This time period would have to be longer in newer domains where suppliers would have to do more design work. Alternately, where the expense or perceived risk to suppliers might be great for designing a software architecture from scratch, a "down select" acquisition strategy could be used in which a limited number of suppliers are chosen to develop competing software architectures.

---

[6] If the expertise for generating and specifying this information does not exist within the acquisition program, a separate contract might be let for this. For a discussion of general scenarios that address particular quality attributes, see the report by Bass and Moreno [Bass 01].

[7] Some organizations refer to a "bidders' conference" as "industry day" briefings to potential suppliers.

[8] The tutorial cannot address any questions about the RFP itself, since only the Contracting Officer is authorized to answer such questions.

Step 3. Conduct ATAM evaluation.

In this step, each potential supplier will conduct its own ATAM evaluation on its own software architecture. Because of the separation between the acquirer and potential suppliers during source selection, certain stakeholder roles (e.g., end users) would need to be represented by surrogates. If a supplier has more skilled and experienced surrogates (e.g., people formerly employed as end users of similar systems), a better evaluation would be expected. The acquirer might wish to engage an impartial third party to observe the ATAM evaluations to ensure that they are performed correctly.

After conducting the ATAM evaluation, the supplier would produce a report that documents the evaluation results along with any other findings and proposed mitigation strategies that will be presented at the subsequent demonstration (see Step 4). The report should be included as part of the technical proposal so that the acquirer can review and evaluate all aspects of the software architecture evaluation's results prior to the demonstration. It is important to specify the appropriate level of detail for the software architecture evaluation's results and subsequent demonstration. One way to address this requirement is by carefully specifying the requirements for documenting the software architecture. Clements and associates discuss the appropriateness of different documentation views to achieve different purposes [Clements 02b]. The evaluation criteria should also help offerors understand the level and type of analysis expected for the demonstration.

Step 4. Conduct ATAM demonstration.

As part of the evaluation process for source selection, all offerors would be required to demonstrate the execution and results of their ATAM evaluations for the acquirer. During this step, the acquirer would ask questions to identify risks, issues, clarifications and deficiencies that the supplier would have to address.

The demonstrations must be conducted in accordance with the source selection policies to ensure fair evaluation of the supplier and to ensure that no "leveling" of suppliers occurs during the demonstration. In addition, it may be appropriate to specify that suppliers may present extensions to the basic information in both the report and demonstration to show additional capabilities.

Step 5. Issue clarifications and deficiencies.

In this step, the acquirer will issue a report to each supplier identifying "clarifications and deficiencies" that are required based on their ATAM demonstrations.

Step 6. Develop final report.

Each potential supplier will respond to the clarifications and deficiencies from their demonstration in a final report. The source selection can then take place using the final reports as inputs to the remainder of the evaluation process for source selection.

We illustrate the application of these steps in the following section. The appendices contain sample RFP language to support this approach.

Step 7. Analyze proposed software architectures and ATAM evaluation results.

In this step, the acquirer will perform an analysis of both the software architecture and ATAM evaluation results for each offeror. The purpose of this step is to rate the proposed software architectures according to their ability to meet the acquisition requirements and to rate the offeror's capability to perform an ATAM evaluation properly and thoroughly.

Examples of proposal contents that would earn poor ratings in this area would include a software architecture that has obvious flaws with respect to achieving key business/mission goals and ATAM evaluation results that overlooked or "whitewashed" important risks and tradeoffs. Examples of proposal contents that would earn good ratings in this area would include proposing a software architecture that is clearly demonstrated to support achievement of key business/mission goals, providing ATAM evaluation results that are thorough and true to the method, identifying important risks (and mitigations), and highlighting key tradeoffs and good design decisions (non-risks). The offeror may also demonstrate modifications to the software architecture based on results of their ATAM evaluation.

If the acquisition office does not possess the technical expertise necessary for this step, an independent third party should be engaged to perform this step and provide the results to the acquirer. This step can serve as a less intrusive means of determining the capability of the offerors to conduct ATAM evaluations than by having an observer present during the ATAM evaluations as described in Step 3.

4 Using the ATAM in a Source Selection: An Example

As we have indicated, there are many ways to incorporate software architecture evaluations into an acquisition. In this section, we have selected one approach to illustrate applying the ATAM in the pre-award and award phases of an acquisition, specifically during source selection.

Examples of appropriate RFP language that correspond to this selected approach are given in the appendices.[9] It must be remembered that software architecture requirements and design contribute substantially to the achievement of system requirements. Therefore, the language in the appendices must be viewed as only a portion of the RFP language and must be integrated into the overall context of the source selection process and the subsequent system acquisition. If a software-only system is being acquired, developing the appropriate language is much easier; this type of acquisition is not the norm, however, in the DoD and government environment.

It is critical to understand that each section of the RFP must be consistent with all the other sections. Sections L and M both affect source selection and must be consistent with Section C. Sections L and M cannot be viewed alone, but must be crafted to align with the context of the entire system acquisition.

4.1 Example Software Architecture Evaluation Approach for Source Selection

The example we discuss here follows a normal source selection process in a competitive environment. In this process, the acquirer prepares a solicitation package containing all necessary sections of an RFP. Sections C, L, and M specifically solicit responses from offerors. Once the responses are received by the acquirer, they are evaluated in accordance with a source selection plan and the criteria described in Section M. In addition to this typical process, we will include language to integrate a demonstration in which each potential supplier presents the ATAM resulting from its analysis of its proposed software architecture as part of the source selection process. As previously discussed, such an approach and the demonstration within the source selection process can be required only if there is reasonable assurance that all offerors already have a software architecture that they can propose for the

[9] Every acquisition is considered unique. The acquisition language provided in this technical note should not be applied directly to all acquisitions. It is strongly recommended that the language be tailored to the acquirer's specific needs.

system being acquired or that sufficient time is allowed for development of suitable software architectures. (See the discussion in Section 3.2.)

For our example and as noted earlier, the acquirer is expected to develop and include the following in the solicitation package:

- preliminary business drivers motivating the development effort
- key quality attributes desired in the system reflecting the business drivers
- a set of scenarios that characterize the quality attributes in operational terms
- requirements for documenting the software architecture to permit analysis of how it supports the required quality attributes

The business drivers correspond to the business goals that are motivating the system acquisition and, in turn, will drive the specification of the system's quality attributes. Examples of business drivers include

- time to deploy
- ability to accommodate operational upgrades
- ability to integrate new subsystems
- ability to interoperate with a specific operational legacy system
- reduced maintenance costs (i.e., ease of modifiability)
- specific performance requirements
- availability or reliability requirements
- security requirements

4.2 Example of RFP/Contract Language for an Acquisition

For our example, we describe language that typically is included in Sections L and M of the RFP. Note again that this language must be consistent with the remainder of the solicitation package requirements such as Section C requirements.

Section L

Section L in our example describes the requirements of the acquisition that each offeror must respond to in their proposal. Consistent with the remainder of the RFP and Section M, software architecture and software architecture evaluations are considered as a part of the overall system acquisition in this example. Key to this section is the required response to the source selection demonstration. The response requires each offeror's plan to perform an ATAM (as discussed in Section 3.1) as part of the source selection demonstration. In addition, Section L asks each offeror for its past performance in the areas of interest for the acquisition.

In our case, this section asks for experience in software architecture design and software architecture evaluation.

Section M

Section M in our example describes how the responses to the RFP from offerors will be evaluated. Here the proposed approach to the software architecture requirements is one of the subfactors under the Technical Factor. The criteria defined in Section M will be applied to this factor and subfactor; namely, the criteria defined in Section M include adequacy of the offeror's response, feasibility of the offeror's approach to satisfy the acquisition requirements, and flexibility of the offeror's approach. In this case, we define flexibility as the extent to which the offeror's approach is adaptable to changing needs or requirements, including future growth.

Finally, the example uses the results of the offeror's evaluation and source selection demonstration to verify the feasibility and flexibility of the proposed approaches and claimed capabilities including the offeror's capability to design and evaluate software architectures. The source selection demonstration requires the offerors to walk through the ATAM and present the ATAM results (again, as discussed in Section 3.1).

5 Summary

In this technical note, we have discussed how a software architecture evaluation, specifically an ATAM-based evaluation, might be used to reduce risk in a system acquisition. We have given examples of RFP language that may be adapted for a particular acquisition. Used appropriately, we believe that this approach will help the acquirer select a supplier that is more capable than other suppliers of developing a software-intensive system that meets its quality goals.

Feedback and Contact

Comments or suggestions about this document or the series of technical notes on software architecture evaluation in the DoD are welcome. We want this series to be responsive to the needs of DoD and government personnel. To that end, comments concerning this technical note, inclusion of other topics, or any other issues or concerns will be of great value in continuing this series. Comments or suggestions should be sent to

Linda Northrop, Director
Product Line Systems Program
lmn@sei.cmu.edu

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Appendix A. Example Section L: Proposal Preparation Instructions (PPI)

The language in this section represents one possible way of requesting areas related to software architecture and software architecture evaluation requirements for the source selection. The language here must be adapted for each acquisition, especially in regulations, policies, and guidance for source selection and for procurement actions. The language shown illustrates that software architecture and software architecture evaluation are in a broader context of a system acquisition. For convenience, the areas applicable to software architecture and software architecture evaluation are in **bold** text. Angle brackets "< >" denote specific information that must be inserted.

Section L

1. General

The offeror's proposal shall be submitted in several volumes as set forth below.

Volume	Title	Copies
Volume 1	Executive Summary	10
Volume 2	Price with Annex	
Volume 3	**Technical**	**10**
Volume 4	**Past Performance**	**7**
Volume 5	**Source Selection Demonstration Plan/Procedures**	**5**
.....		

1.2 All information pertaining to a specific volume shall be confined to that volume.

1.3 The offeror shall completely describe the approach, with supporting rationale, to complete each Statement of Work (SOW) task or meet each RFP requirement as specified in this PPI. The offeror shall provide sufficient details and substantive information to convey to the evaluator a clear and accurate understanding of how the requirement is to be met and to permit a complete and accurate evaluation of the

proposal. The offeror shall identify all risks, uncertainties, or major problems in meeting the technical, delivery, quality objectives, and other requirements of the RFP. The proposed mitigation of these risks, uncertainties, or resolution of the problems will be provided.

2. PROPOSAL FORMAT AND CONTENT

VOLUME 1 - EXECUTIVE SUMMARY

VOLUME 2 - PRICE WITH ANNEX

VOLUME 3 - TECHNICAL

This Volume shall contain a full discussion of how the offer and proposed approach intends to satisfy requirements identified in the respective paragraphs of the RFP.

Software Architecture

For each paragraph associated with the software architecture in the Statement of Work (SOW), the offeror shall describe the proposed software architecture, the approach to the development and evaluation of the final software architecture and how this approach will result in a software architecture to meet the RFP requirements.

VOLUME 4 - PERFORMANCE RISK

Volume 4 shall contain a full discussion of how the offeror intends to satisfy the RFP requirements indicated below.

Volume 4 will be partitioned as follows:

Past Performance
Management Control Environment
Organization
Project Management
Data Management
Schedule
Facilities

The contents of these Sections are defined as follows:

4.1 Past Performance

Describe work performed on software projects similar in scope to the requirements for <SYSTEM NAME>, to include design methodology, software architecture design, software architecture evaluation, software integration, integration of NDI software, utilization of industry standards for developing and integrating software (e.g., open

system architecture), software security, computer aided software engineering (CASE) tools, and original estimated lines of code versus actual lines of code at completion.

Discuss engineering management techniques utilized on similar efforts to control cost, schedule, and performance. All efforts to mitigate risks, along with the degree of success of the actions taken to mitigate risk, must be provided.

Discuss concurrent engineering approaches, including software architecture development and evaluation that were used, including lessons learned, and resulting engineering, manufacturing, and equipment improvements that enhanced equipment and contract performance.

In the event that the offeror/subcontractor(s) (if applicable) has not had applicable contracts, a summary of other experience with similar and/or related work over a like period of time shall be submitted with POCs for each customer.

The Government may elect to verify all or some past performance data provided in the proposal by obtaining additional information outside of the written content of the proposal. In addition, the Government may consider relevant data extrinsic to the proposal which is otherwise available to the Government. In the event of an unresolved discrepancy, the Government-obtained data will take precedence.

VOLUME 5 - SOURCE SELECTION DEMONSTRATION PLANS/PROCEDURES

The Government intends, through Source Selection Demonstrations, to verify the capabilities of the proposed hardware and software items and associated software architectures. Results of the Source Selection Demonstrations will be used to verify the feasibility and flexibility of the proposed approaches and claimed capabilities to satisfy the <SYSTEM NAME > requirements. The demonstrations must be sufficient to verify the proposed approaches and claimed capabilities. The offerors shall conduct demonstrations using existing hardware/software. It is not the Government's intent to burden the offerors with development of <SYSTEM NAME > unique hardware/software for the purposes of this demonstration.

Demonstrations will take place at the Government facilities at <LOCATION>. The demonstration is solely an offeror demonstration with Government representatives observing. The Government may query the offeror during the demonstration regarding the proposed capability being demonstrated or regarding the plans and procedures being performed.

Volume 5 of the proposal will contain the Source Selection Demonstration plan and procedures, which will be used by the offeror during the conduct of the demonstration. The plan and procedures will be developed using as a guide for format and content <ACQUIRER'S STANDARD TEMPLATES>.

The plan and procedures will address all demonstrations and their sequence, and specific schedules of events for each demonstration, as defined in this section of the RFP. The demonstration schedule shall be in a matrix format as shown by the sample below. Offeror will not be allowed to conduct simultaneous demonstrations. The demonstration plan and procedures are considered part of the proposal and as such, the

Government will assess the plan and procedures. The Government may forward comments to the offeror based upon such assessments. The plan and procedures submitted in this volume, as modified as a result of Government comments, will be the only ones used by the Government and the offeror during the demonstrations. Any deviations or changes to these plans and procedures will require the offeror during his scheduled demonstration period to review, in detail with Government observers, the reason for the deviation/change and explain how that deviation/change is necessary to verify the capability being demonstrated. This review shall be conducted prior to the demonstration involving the deviation/change.

The offeror will be allotted four (4) weeks, for a total of 140 hours, in which to conduct and complete his demonstration of the system. The offeror may demonstrate other unique capabilities in addition to the "SOW Requirements to be Demonstrated" within the allotted total time. The offeror shall allocate time for unique demonstrations and re-demonstrations within this time frame. However, re-demonstrations will be performed within the time frame for the specific equipment/software category given in this Section (see sample schedule below). The hours of demonstration will be 0800-1130 and 1230-1600 Monday-Friday.

Offeror must bring sufficient equipment and other material, e.g., documentation, to accomplish demonstrations, as well as spares in the event of equipment failures. Offerors are completely responsible for the physical control and maintenance of their equipment.

The Government also intends to conduct an audit of all offeror equipment and software to be demonstrated or used in the demonstration. The offeror shall be allowed twelve (12) hours to set up all his equipment/software and to conduct the audit. It is planned that the set up and audit will commence at 0800 hours one day prior to the scheduled start date of the demonstrations, to allow maximum time for demonstrations. The set up and audit will be completed by 2000 hours on the day started. If additional time is needed by the offeror, it will be completed before the demonstrations are started and this additional time, if required, shall be subtracted from the offerors' allowed 140 hours for conducting all of the demonstrations.

The Government will require the offeror to perform the audit under Government control and direction, including opening the hardware for Government inspection and identifying software. No changes or modification to the equipment or software will be allowed after the audit without Government approval. The Government reserves the right to revalidate the audit or conduct additional audits, as necessary, during the demonstration period.

For the purposes of the demonstration, the requirements to be demonstrated are those stated in the system specification, including those requirements related to the software architecture and the software architecture evaluation.

For conduct of the demonstration the offeror shall prepare the Source Selection Demonstration plans/procedures to be used in conducting the demonstration. The system capabilities will be demonstrated in the following order:

Weeks one and two:

1. software architecture

2. <Other capability to be demonstrated>
3. <Other capability to be demonstrated>

Weeks three and four:

4. <Other capability to be demonstrated>
5. <Other capability to be demonstrated>

•
•

For the software architecture and software architecture evaluation portion of the demonstration, the offeror shall conduct an ATAM prior to submission of proposal following the evaluation steps described in the Attachment A: ATAM Evaluation Steps to this PPI. The offeror must use scenarios provided by the Government in the RFP as part of this ATAM.

The offeror must designate a Demonstration Director who will be the sole responsible person to interface with the Government-appointed Demonstration Director/Leader during the conduct of the demonstration. The offeror's designee must be identified prior to the demonstration and must be available during the entire demonstration.

To the extent that the software and associated software architecture to be demonstrated differs from that which is offered for delivery, the offeror must completely describe the differences in this volume. The offeror shall fully describe in this volume his approach to providing the proposed software and associated software architecture meeting the requirements of the demonstration.

No demonstrations will be performed without procedures submitted as part of the proposal.

Attachment A

ATAM Evaluation Steps

For the software architecture and software architecture evaluation portion of the demonstration, the offeror shall conduct an ATAM evaluation prior to submission of proposal following the evaluation steps described in this attachment. The offeror must use the business drivers, quality attributes, and associated scenarios provided by the Government in the RFP as a starting point for this ATAM evaluation. All roles in each of the ATAM steps must be performed by the offeror including any surrogates who represent stakeholders.

There are nine specific steps in the basic ATAM evaluation that fall into four general types of activities, Presentation, Investigation and Analysis, Testing, and Reporting. (See *Evaluating Software Architectures: Methods and Case Studies* by Clements, Kazman, and Klein, published by Addison Wesley, Reading, MA, 2002).

Presentation

Step 1. Present the ATAM. The method is described to the assembled stakeholders (typically customer representatives, the architect or software architecture team, user representatives, maintainers, administrators, managers, testers, integrators, etc.).

Step 2. Present business drivers. The project manager describes the business goals that are motivating the development effort and hence the primary software architecture drivers (e.g., high availability, time to market, high security, etc.).

Step 3. Present software architecture. The architect describes the initial proposed software architecture, focusing on how it addresses the business drivers.

Investigation and Analysis

Step 4. Identify architectural approaches. Software architecture approaches are identified by the architect, but are not analyzed.

Step 5. Generate quality attribute utility tree. The quality attributes that comprise system "utility" (performance, reliability, security, modifiability, etc.) are elicited from the assembled stakeholders. These are specified down to the level of scenarios, annotated with stimuli and responses, and prioritized. (A scenario is a short statement describing an interaction of a stakeholder with the system. Scenarios provide a vehicle for making vague qualities concrete.)

Step 6. Analyze architectural approaches. Based upon the high-priority factors identified in Step 5, the architectural approaches that address those factors are elicited and analyzed. For example, an architectural approach aimed at meeting performance goals will be subjected to a performance analysis. During this step, software architecture risks, sensitivity points, and tradeoff points are identified.

Testing

Step 7. Brainstorm and prioritize scenarios. Based upon the example scenarios generated in the utility tree step, a larger set of scenarios is elicited from the entire group of stakeholders. This set of scenarios is prioritized via a voting process involving the entire stakeholder group.

Step 8. Analyze architectural approaches. This step reiterates Step 6; but here, the highly ranked scenarios from Step 7 are considered to be test cases for software architecture approaches determined thus far. These test case scenarios may uncover additional software architecture approaches, risks, sensitivity points, and tradeoff points that are then documented.

Reporting

Step 9. Present results. Based upon the information collected during the ATAM evaluation (styles, scenarios, attribute-specific questions, the utility tree, risks, sensitivity points, tradeoffs), the ATAM evaluation team presents its findings to the assembled stakeholders and details this information along with any proposed mitigation strategies in a written report.

For the Demonstration, the offeror will accomplish steps 1 to 9 prior to the demonstration and prior to generating the demonstration plan and procedures. The report of the results (Step 9) shall be submitted by the offeror as part of the proposal. The report will describe how the offeror accomplished each step of the ATAM and associated results of the ATAM evaluation. This must include identified sensitivity points, tradeoffs, risks and "non-risks" (good design decisions that rely on assumptions that are frequently implicit in the software architecture).

During the demonstration, the offeror shall report to the Government the results of each step of the ATAM. The following sample agenda will be part of the offeror's demonstration plan.

Sample Agenda
Day 1

Time	Agenda item
0830-0900	Present the ATAM approach
0900-1000	Present Business Drivers: Review business drivers, quality attributes, scenarios used in the ATAM
1000-1030	Break

1030-1200	Present initial software architecture
1200-1300	Lunch
1300-1430	Present results of analysis
1430 -1500	Break
1500-1630	Present results of analysis

Day 2

Time	Agenda item
0830-1000	Present proposed software architecture resulting from addressing results of ATAM evaluation
1000-1030	Break
1030-1200	Analyze Government provided scenario interaction with proposed software architecture
1200-1300	Lunch
1300-1430	Analyze scenario interaction*
1430-1530	Wrap-up; summarizing issues; reporting next steps

*In this step, the offeror will use the Government provided business drivers, quality attributes, and scenarios to demonstrate how the proposed software architecture satisfies these requirements.

The offeror shall document in a report the results of this presentation including identification of issues and risks found during the presentation.

Appendix B. Example Section M: Basis for Award

The language in this section represents one possible solution to evaluating responses or proposals for each offer's software architecture approach to satisfy the requirements of the acquisition. The language here must be adapted for each acquisition, especially in light of each acquisition organization's regulations, policies, and guidance for source selection and for procurement actions.

The language shown attempts to illustrate that software architecture and software architecture evaluation is in a broader context of a system acquisition. For convenience, the areas applicable to software architecture and software architecture evaluation are in bolded text.

Section M

1. Basis For Award

The award of the <SYSTEM NAME> contract will be based upon the offer that provides the best overall value to the Government in terms of technical, prices, [and] performance risk. All proposals will be evaluated in terms of the factors and subfactors in accordance with the criteria set forth below. Award may not necessarily be made to the offeror with the lowest evaluated price.

2. Factors And Subfactors To Be Evaluated

The following factors and subfactors will be evaluated.

FACTOR:	**Technical** **Subfactors:** **Hardware** **Software architecture** **Software**
FACTOR:	Price
FACTOR:	**Performance Risk**
FACTOR:	Management

3. Relative Importance of Evaluation Factors/Subfactors

The Technical and Price factors are equal in importance. The Technical and Price factors combined are significantly more important than the other factors combined.

4. Evaluation Criteria

The following criteria will be applied to measure the quality of the proposed approach under the Technical, and Performance Risk factors and their respective subfactors, as indicated in Paragraph 5 below.

4.1 Adequacy of Response

Adequacy of response is defined as the extent to which the proposed approach is complete and demonstrates an understanding of the requirements.

Completeness is defined as the extent to which: the proposal describes approaches, including proposed solutions, that address all requirements of the acquisition as requested in the RFP, Section L, and associated risks; means for resolution of the risks have been provided; and the approaches are discussed with sufficient, substantive information to convey to the evaluator a clear and accurate description of how the requirements are to be satisfied.

Understanding of requirements is the extent to which the approach, including proposed solutions, demonstrates an accurate comprehension of the specified requirements, the intended mission environment, and program goals.

4.2 Feasibility

Feasibility is defined as the extent to which the approach, including proposed solutions, is capable of satisfying requirements and is realistically achievable, including the extent to which all risks associated with the approach have been mitigated for successful achievement of the requirements.

4.3 Flexibility

Flexibility is the extent to which the approach is adaptable to changing needs or requirements, including future growth. For evaluation of software architecture, flexibility is further defined in terms of modifiability, security, and reusability, which are defined as:

Modifiability - the extent to which the system can be changed quickly and cost effectively

Reliability - a measure of the proportion of time the system is up and running.

Security - a measure of the system's ability to resist unauthorized attempts at usage and denial of service, while still providing its services to legitimate users.

4.4 Performance Risk Assurance

Performance Risk Assurance (PRA) is defined as the Government's level of confidence that the offeror (including each subcontractor/team member) will meet technical, delivery, quality, and small disadvantaged business subcontracting objectives of the <SYSTEM NAME> contract, based upon the degree that the offeror (including each subcontractor/ team member) has met these same objectives for similar and related

efforts, and based upon the feasibility of his proposed management and technical approaches for the <SYSTEM NAME > contract.

4.5 Source Selection Demonstration

Results of the Source Selection Demonstration will be used to verify the feasibility and flexibility of the proposed approaches and claimed capabilities to satisfy the <SYSTEM NAME > requirements, including the offeror's capability to design and evaluate software architectures, and the offeror's understanding of the requirements.

5. Evaluation Approach

5.1 FACTOR: Technical

The Technical factor will be evaluated in terms of its adequacy of response, feasibility, and flexibility.

5.2 FACTOR: Price

5.3 FACTOR: Performance Risk

The Performance Risk factor will be evaluated in terms of performance risk assurance.

5.4 FACTOR: Management

Appendix C. Example Quality Attributes and Scenarios

Quality Attributes

The following quality attributes for the system are derived from the business drivers for the program.

The software architecture will ensure achievement of system functions and the quality requirements of modifiability, reliability, and security, as described in this specification.

Quality attributes for the software are defined as:

Modifiability	The extent to which the system can be changed quickly and cost-effectively.
Reliability	A measure of the proportion of time the system is up and running.
Security	A measure of the system's ability to resist unauthorized attempts at usage and denial of service, while still providing its services to legitimate users.

Potential Evaluation Scenarios

A *scenario* is a short statement describing an interaction of one of the system stakeholders with a system. A scenario must consist of three parts: stimulus, environment, and response. The stimulus describes what the stakeholder does to initiate the interaction with the system. The environment describes what is going on at the time of the stimulus. The response tells how the system should respond to the stimulus. A complete set of scenarios must include use case scenarios (typical uses of the system), growth scenarios (anticipated changes to the system), and exploratory scenarios (extreme changes to the system that "stress" the system). (See *Evaluating Software Architectures: Methods and Case Studies* by Clements, Kazman, and Klein, published by Addison Wesley, Reading, MA, 2002).

The software architecture will also ensure achievement of the system quality attributes shown in the following potential evaluation scenarios:

Quality Attribute	Potential Evaluation Scenarios
Modifiability	Changes to the output data format are possible with one person-week of effort. ...
Reliability	Target information overloads the system with 50 targets simultaneously. The system notifies the operator of the overload but continues to process as many targets as possible within one second while operating 99.9% of the time. ...
Security	The system is able automatically to prevent unauthorized entry attempts through the communication system connected to an external client and log data to assist in tracing the source of the attempt. ...

Glossary

Functionality — The ability of the system to do the work for which it was intended

Modifiability — The extent to which the system can be changed quickly and cost-effectively

Performance — The responsiveness of the system – the time required to respond to stimuli (events), or the number of events processed in some interval of time

Reliability — A measure of the proportion of time the system is up and running

Scenario — A brief description of a stakeholder's interaction with a system; how a system behaves or interacts with the stakeholders to accomplish desired objectives or stated requirements

Security — A measure of the system's ability to resist unauthorized attempts at usage and denial of service, while still providing its services to legitimate users

Software Architecture — The system and the structure or structures of the system, which include software components, the externally visible properties of those components, and the relationships among them

References

[Bass 98] Bass, L.; Clements, P.; & Kazman, R. *Software Architecture in Practice.* Reading, MA: Addison-Wesley Publishing Co., 1998.

[Bass 01] Bass, L.; Klein, M.; & Moreno, G. *Applicability of General Scenarios to the Architecture Tradeoff Analysis Method* (CMU/SEI-2001-TR-014, ADA 396098). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, October 2001. <http://www.sei.cmu.edu/publications /documents/01.reports/01tr014.html>.

[Bergey 99] Bergey, J.; Fisher, M.; & Jones, L. *The DoD Acquisition Environment and Software Product Lines* (CMU/SEI-99-TN-004, ADA 244787). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, May 1999. <http://www.sei.cmu.edu/publications/documents/99.reports /99tn004/99tn004abstract.html>.

[Bergey 01] Bergey, J. & Fisher, M. *Use of the Architecture Tradeoff Analysis Method (ATAM) in the Acquisition of Software Intensive Systems*, (CMU/SEI-2001-TN-009, ADA 396096). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, September 2001. <http://www.sei.cmu.edu /publications/documents/01.reports/01tn009.html>.

[Clements 02a] Clements, P.; Kazman, R.; & Klein, M. *Evaluating Software Architectures: Methods and Case Studies*. Reading, MA: Addison-Wesley Publishing Co., 2002.

[Clements 02b] Clements, P. et al. *Documenting Software Architectures: Views and Beyond*. Reading, MA: Addison-Wesley Publishing Co., 2002.

[Cooper 02] Cooper, J. & Fisher, M. *Software Acquisition Capability Maturity Model (SA-CMM), Version 1.03* (CMU/SEI-2002-TR-010, ADA 3399794). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, March 2002. <http://www.sei.cmu.edu/publications/documents /02.reports/02tr010.html>.

[Kazman 00] Kazman, R.; Klein, M.; & Clements, P. *ATAM: Method for Architecture Evaluation,* (CMU/SEI-2000-TR-004, ADA386629). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, August 2000. <http://www.sei.cmu.edu/publications/documents/00.reports/00tr004.html>.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	June 2002	Final

4. TITLE AND SUBTITLE
Use of the Architecture Tradeoff Analysis Method℠ (ATAM℠) in Source Selection of Software-Intensive Systems

5. FUNDING NUMBERS
F19628-00-C-0003

6. AUTHOR(S)
John K. Bergey, Matthew J. Fisher, Lawrence G. Jones

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER
CMU/SEI-2002-TN-010

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
HQ ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2116

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified/Unlimited, DTIC, NTIS

12B DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)
Software architecture is critical to the quality of a software-intensive system. For an acquisition organization, such as the Department of Defense (DoD), the ability to evaluate software architectures as early as possible in an acquisition can have a favorable impact on the delivered system. This technical note explains the role of software architecture evaluation in a source selection and describes the contractual elements that are needed to support its use. The note then briefly describes the Architecture Tradeoff Analysis Method℠ (ATAM℠) and provides an example that shows how to apply this method in a source selection. The example includes sample contractual language that an acquirer can adapt to meet specific acquisition needs.

14. SUBJECT TERMS
Architecture Tradeoff Analysis Method (ATAM), contract language, request for proposal, software architecture, system acquisition

15. NUMBER OF PAGES
44

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL